

はやわかり L^AT_EX で数式組版

Michael Downes, updated by Barbara Beeton

アメリカ数学会

このドキュメントのバージョン 2.0 (2017/12/22) です。配布元は
<https://www.ams.org/tex/amslatex>
です。日本語訳は 2018-09-04 です。

Contents

1	イントロダクション	3
2	文章中の数式（インライン数式）と数式だけの行（ディスプレイ数式）	3
2.1	基本	3
2.2	自動番号付けと相互参照	5
3	数学記号と数式フォント	6
3.1	数学記号の分類	6
3.2	意図的にここで省略された記号	6
3.3	アルファベットと数	7
3.3.1	ラテン文字とアラビア数字	7
3.3.2	ギリシア文字	7
3.3.3	その他の“基本”アルファベット記号	7
3.3.4	数式フォントの変更	8
3.3.5	黒板太字 (msbm; ただし小文字はありません)	8
3.3.6	飾り文字 (cmsy; 小文字はありません)	8
3.3.7	CM フォントではない飾り文字と手書き文字	9
3.3.8	フラクトゥール文字 (eufm)	9
3.4	いくつかの単純な記号	9
3.5	二項演算の記号	10
3.6	関係記号: \lt \leq \gt \sim および, それらの変種	10
3.7	関係記号: 矢印	11
3.8	関係記号: その他	11
3.9	大きさの変わる作用素	12
3.10	句読点	12
3.11	ペアになる (大きさの変わる) 記号	12
3.12	ペアでは使われないが大きさの変わる記号	12
3.13	大きさの変わる垂直の矢印	12
3.14	数式で使われるアクセント	13
3.15	名前付き作用素	13
4	記法	13
4.1	上側と下側を飾る	13
4.2	長さが伸びる矢印	13
4.3	記号に別の記号を重ねる	14
4.4	行列	14
4.5	数学用の空白コマンド	14
4.6	ドット	15
4.7	分割させないダッシュ	15
4.8	根号	15
4.9	数式の枠囲み	15

5	分数とそれに関連する話題	16
5.1	<code>\frac</code> , <code>\dfrac</code> , および <code>\tfrac</code> コマンド	16
5.2	<code>\binom</code> , <code>\dbinom</code> , および <code>\tbinom</code> コマンド	16
5.3	<code>\genfrac</code> コマンド	16
5.4	連分数	17
6	デリミタ	17
6.1	デリミタの大きさ	17
6.2	縦棒の記法	18
7	<code>\text</code> コマンド	18
7.1	<code>\mod</code> と, その仲間	18
8	積分と和	18
8.1	範囲指定の配置	18
8.2	多重積分	19
8.3	複数行の下付き文字または上付き文字	19
8.4	<code>\sideset</code> コマンド	19
9	数式の構成部分の大きさを変える	19
10	このほかのパッケージ	20
11	その他の役に立つドキュメント	21

謝辞と将来の計画

提案, 援助, 励ましなど貢献して下さったすべての人に感謝します. 難しいマクロを修復した David Carlisle と AMS スタイルにしたがうように編集してくれた Jennifer Wright Sharp に特別に感謝します.

将来の版の計画には, 索引の追加が含まれます.

エラーに関する報告と改善のための提案は, tech-support@ams.org に送ってください.

1. イントロダクション

これは L^AT_EX と数式のあるドキュメントを作成するためのいくつかの拡張パッケージの、お勧めの機能についての簡潔な要約です。詳細な説明に関心がある読者は、参考文献 (Reference) の、特に [Lam], [AMUG], そして [LFG] をみてください。普通の L^AT_EX については、ある程度の経験があることを想定しています。L^AT_EX が使う、コマンド、オプションの引数、環境、パッケージなどを思い出す必要があれば、[Lam] を参照してください。

ここで説明している機能のほとんどは、L^AT_EX をアメリカ数学会が作成した 2 つの拡張パッケージ `amssymb` と `amsmath` を読み込んで使用すれば、利用できます。したがって、このドキュメントのソースファイルは

```
\documentclass{article}
\usepackage{amssymb,amsmath}
```

で始まります。数式や記号の使用量が比較的少ないドキュメントの場合、`amssymb` パッケージは使わなくても良いでしょう。3 節では、`amssymb` を必要とする記号に ^a または ^b (それぞれ `msam` フォントまたは `msbm` フォント) とマークされます。3.3 節には、いくつかの追加フォントが含まれています。必要なパッケージがそこで示されています。

その他の便利な機能を提供するパッケージもありますが、このドキュメントでは扱いません。それらについては 10 節を参照してください。数学記号については、特にここで紹介したリストは、すべてではなく、ユーザーの L^AT_EX システムに普通に備わっているものです。フォントを使うためにインストールしたり他の設定をしなくても使用できるような記号だけを扱っています。

ここに示されている記号以外が必要な読者は、*The Comprehensive L^AT_EX Symbol List* [CLSL] を参考にするとよいでしょう。お使いの L^AT_EX が T_EX Live に基づいており、ドキュメントも含まれていれば、システムプロンプトで `texdoc comprehensive` と入力することで、記号一覧を読むことができます。

2. 文章中の数式 (インライン数式) と数式だけの行 (ディスプレイ数式)

2.1. 基本. L^AT_EX で数式作成モードに入ったり出たりするのは、通常以下のコマンドと環境で行われます。

インライン数式	ディスプレイ数式
$\$ \dots \$$	<code>\[...\]</code> 数式番号がつかない
$\left(\dots \right)$	<code>\begin{equation*}</code> 数式番号がつかない
	<code>\end{equation*}</code>
	<code>\begin{equation}</code> 自動的に数式番号がつけられる
	<code>\end{equation}</code>

注意 1. 本文とディスプレイ数式の間には空白行を入れないでください。そうしてしまうと、その場所での改ページが可能になります。これは悪いスタイルです。また、文章とディスプレイ数式の間隔が不正確になり、通常は望ましい値よりも大きくなります。ドキュメント作成時の見やすさのために改行が必要な場合は、先頭に % だけの行を挿入します。新しい段落が意図されている場合にのみ、ディスプレイ数式の後と次のテキストとの間に空白行を置きます。

注意 2. 入力 (`\[...\]`, `equation` など) に複数のディスプレイ数式をグループ化しないでください。その代わりに、(`split`, `aligned` など) などの複数行構造のためのコマンドを適切に使用してください。

注意 3. 実際には、`\begin{math}` と `\begin{displaymath}` の両方が必要になることはめったにありません。ディスプレイ数式の作成に plain T_EX 表記 `$$ \dots $$` を使用することは避けてください。L^AT_EX では明示的に禁止されていませんが、L^AT_EX コマンドの一部として L^AT_EX の本のどこにも説明されておらず、L^AT_EX コマンドの一部としてドキュメント化されていません。これはたとえば `fleqn` オプションを使用するとき特別な問題をおこします。

注意 4. [Lam] に記述されている `eqnarray` 環境と `eqnarray*` 環境の使用も避けてください。これを使うと等号の間隔が不均一になり、式番号と方程式本体が重なるのを防げません。

方程式のグループ化と複数行を扱うための環境は、Table 1 に示されています。

Table 1: 複数の行が必要な数式と数式のグループ
(縦棒は, 通常のマージンを示しています).

<pre>\begin{equation}\label{xx} \begin{split} a& =b+c-d\\ & \quad +e-f\\ & =g+h\\ & =i \end{split} \end{equation}</pre>	$\left. \begin{aligned} a &= b + c - d \\ &+ e - f \\ &= g + h \\ &= i \end{aligned} \right \quad (1.1)$
<pre>\begin{multline} a+b+c+d+e+f\\ +i+j+k+l+m+n\\ +o+p+q+r+s \end{multline}</pre>	$\left. \begin{aligned} a + b + c + d + e + f \\ + i + j + k + l + m + n \\ + o + p + q + r + s \end{aligned} \right \quad (1.2)$
<pre>\begin{gather} a_1=b_1+c_1\\ a_2=b_2+c_2-d_2+e_2 \end{gather}</pre>	$\left. \begin{aligned} a_1 &= b_1 + c_1 & (1.3) \\ a_2 &= b_2 + c_2 - d_2 + e_2 & (1.4) \end{aligned} \right $
<pre>\begin{align} a_1& =b_1+c_1\\ a_2& =b_2+c_2-d_2+e_2 \end{align}</pre>	$\left. \begin{aligned} a_1 &= b_1 + c_1 & (1.5) \\ a_2 &= b_2 + c_2 - d_2 + e_2 & (1.6) \end{aligned} \right $
<pre>\begin{align} a_{11}& =b_{11}& \\ & a_{12}& =b_{12} \\ a_{21}& =b_{21}& \\ & a_{22}& =b_{22}+c_{22} \end{align}</pre>	$\left. \begin{aligned} a_{11} &= b_{11} & a_{12} &= b_{12} & (1.7) \\ a_{21} &= b_{21} & a_{22} &= b_{22} + c_{22} & (1.8) \end{aligned} \right $
<pre>\begin{alignat}{2} a_1& =b_1+c_1 & & & +e_1-f_1 \\ a_2& =b_2+c_2 & -d_2 & +e_2 \end{alignat}</pre>	$\left. \begin{aligned} a_1 &= b_1 + c_1 & + e_1 - f_1 & (1.9) \\ a_2 &= b_2 + c_2 - d_2 + e_2 & (1.10) \end{aligned} \right $
<pre>\begin{flalign} a_{11}& =b_{11}& \\ & a_{12}& =b_{12} \\ a_{21}& =b_{21}& \\ & a_{22}& =b_{22}+c_{22} \end{flalign}</pre>	$\left. \begin{aligned} a_{11} &= b_{11} & a_{12} &= b_{12} & (1.11) \\ a_{21} &= b_{21} & a_{22} &= b_{22} + c_{22} & (1.12) \end{aligned} \right $

注意 1. *を任意のプライマリ環境に適用すると, 数式番号の割り当てが抑止されます. ただし, `\tag` はラベルを表示するために, `\eqref` は手動でタグ付けされた行を参照するために使用できます. 従属環境で*または`\tag`を使用するとエラーになります.

注意 2. `split` 環境は特殊なケースです. これは従属環境で, 複数の式をもつ構造の `equation` 環境の内容や `aligned` や `gather` のような複数式の中の 1 ディスプレイ数式つの “line” の内容として使用できます.

注意 3. 主要な環境 `gather`, `align` と `align` には, その下位構造に “-ed” 対応 (`gather`, `aligned`, `aligned`) があります. より複雑なディスプレイ数式の構成要素として, またはインライン数式の中で使用することができます. これらの “-ed” 環境は, オプションの `[t]`, `[c]` または `[b]` を使用して垂直に配置できます.

注意 4. `flalign` という名前は, “全幅 (full length)” を意味します. よく “左寄せ (flush left)” だと誤解されます. しかし, ディスプレイ数式はしばしば左端から始まり幅全体にわたるので, この混乱は納得できます. 両方のマージンから `flalign` に適用されるインデントは, `\multlinegap` で設定されます.

2.2. 自動番号付けと相互参照. 自動的に数式番号が付く式を作るには, `equation` 環境を使用します. 相互参照のラベルを割り当てるには, `\label` コマンドを使用します.

```
\begin{equation}\label{reio}
...
\end{equation}
```

自動的に番号付けた数式を相互参照するには, 次のようにします. `\eqref` コマンド:

```
... using equations~\eqref{ax1} and~\eqref{bz2}, we
can derive ...
```

上に示した例では,

using equations (3.2) and (3.5), we can derive

のように表示されます. 言い換えれば, `\eqref{ax1}` は (`\ref{ax1}`) と同じですが, `\eqref` で生成されたカッコは常に立体です.

式番号を $m.n$ (節の番号. 数式番号) という形にしたいときは, ドキュメントのプレアンブルで `\numberwithin` コマンドを使います:

```
\numberwithin{equation}{section}
```

番号付けの方針の詳細については, [Lam, §6.3, §C.8.4] を参照してください.

`subequations` 環境は, グループ内の方程式の番号付の方針のための便利な方法を提供します. たとえば, 現在の方程式の番号が 2.0 であるなら,

```
\begin{equation}\label{first}
a=b+c
\end{equation}
この間に文章があるとして,
\begin{subequations}\label{grp}
\begin{align}
a&=b+c\label{second}\\
d&=e+f+g\label{third}\\
h&=i+j\label{fourth}
\end{align}
\end{subequations}
```

とすれば

$$a = b + c \tag{2.1}$$

この間に文章があるとして,

$$a = b + c \tag{2.2a}$$

$$d = e + f + g \tag{2.2b}$$

$$h = i + j \tag{2.2c}$$

となります. `\begin{subequations}` の直後に `\label` コマンドを置くことで, 親の番号への参照を得ることができます. 上記の例の `\eqref{grp}` は (2.2) を生成し, `\eqref{second}` は (2.2a) を生成します.

上で述べた数式番号の付け方のいくつかの変種が <https://tex.stackexchange.com/questions/220001/> に紹介されています. そこには (2.1), (2.2a) ... ではなく (2.1), (2.1a) ... とする例などがあります. これは, `\tag` を使って, 子番号をもつ式の親の式に対して相互参照を行うことで実現されます.

3. 数学記号と数式フォント

3.1. 数学記号の分類. 数式で使う記号は, いくつかの種類 (クラス) に分けられます. 数式が単語で表されるなら, 読み方に多少なりとも対応します. 式の可読性を高めるために, 異なる記号クラスごとに伝統的に間隔および配置が使用されています.

クラス 番号	ニーモニック	意味 (読みのヒント)	例
0	Ord	普通/通常 (名詞)	$A \ 0 \ \Phi \ \infty$
1	Op	前に置かれる作用素	$\sum \ \Pi \ \int$
2	Bin	二項作用素 (結合)	$+ \cup \ \wedge$
3	Rel	関係/比較 (動詞)	$= \ < \ \subset$
4	Open	左/開きのデリミタ	$(\{ \{ \langle$
5	Close	右/閉じのデリミタ	$\rangle \} \})$
6	Punct	後置記号/文の終わり	$., ; !$

注意 1. T_EX のクラス 0 とクラス 7 の区別は, フォント選択の問題にのみ影響します. ここでは重要ではありません.

注意 2. クラス 2 (Bin) の記号, 特にマイナス記号 $-$ は, L^AT_EX によってクラス 0 (空白なし) として, 自動的にプリントされます. つまり適切な左オペランドがないとき, つまり数式の始まりまたは開始デリミタの受け側であるとしています.

いくつかの記号の間隔は, 日常ドキュメントではなく数式の伝統に従います: $/$ は (意味的に) クラス 2 のものですが, $k/2$ ではなくスラッシュのまわりに空白を入れずに $k/2$ と書きます. そして, $p|q \ p|q$ (空白なし) と $p \mid q \ p \mid q$ (クラス 3 の空白) とを比べてみましょう.

新しい数学記号を定義する適切な方法は, L^AT_EX 2_ε フォントの選択で説明されています [LFG]. フォントの仕様の影響を最初に理解する必要があるため, ここでは有用な概要を与えることは実際には不可能です. しかし, wncyr10 という名前のキリル文字フォントが利用できることを知っているとして, そのフォントの 1 文字を数学記号としてプリントするための L^AT_EX コマンドでの定義方法について最小の例を示します:

```
% Declare that the combination of font attributes OT2/wncyr/m/n
% should select the wncyr font.
\DeclareFontShape{OT2}{wncyr}{m}{n}{<->wncyr10}{}
% Declare that the symbolic math font name "cyr" should resolve to
% OT2/wncyr/m/n.
\DeclareSymbolFont{cyr}{OT2}{wncyr}{m}{n}
% Declare that the command \Sh should print symbol 88 from the math font
% "cyr", and that the symbol class is 0 (= alphabetic = Ord).
\DeclareMathSymbol{\Sh}{\mathalpha}{cyr}{88}
```

3.2. 意図的にここで省略された記号. 次に示す数学記号は L^AT_EX book [Lam] に記載されていますが, amssymb パッケージがロードされたときに同等の記号に置き換えられるため, 意図的に省略しています. amssymb パッケージを使用している場合は, 代替名を使用することはたんにドキュメントで使用されるフォントの数を不必要に増やしだけです.

```
\Box, 右のコマンドを使う \square □
\Diamond, 右のコマンドを使う \lozenge ◇
\leadsto, 右のコマンドを使う \rightsquigarrow ↗
\Join, 右のコマンドを使う \bowtie ⋈
\lhd, 右のコマンドを使う \vartriangleleft ◁
\unlhd, 右のコマンドを使う \trianglelefteq ≐
\rhd, 右のコマンドを使う \vartriangleright ▷
\unrhd, 右のコマンドを使う \tranglerighteq ≐
```

さらに, L^AT_EX には, ここに含まれているものよりも強力なものがたくさんあります. このリストは包括的なものではありません. 音標アルファベットや記号のような数学では必要ない記

号も含まれた包括的なリストについては, *The Comprehensive L^AT_EX Symbol List* [CLSL] を参照してください. (すべてのフォントの一覧は, フォント名順に T_EX Live:texdoc rawtables が提供するドキュメントに含まれています. これらの表には記号名は含まれません.) この他の記号についての情報については unicode-math パッケージにあります; [UCM] を参照してください.

3.3. アルファベットと数.

3.3.1. ラテン文字とアラビア数字

ラテン文字は, クラス 0 の単純な記号です. 数式内のラテン文字のデフォルトのフォントはイタリックです.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z

数式で i や j にアクセントを加えるときには, ドットのない i や j は `\imath` や `\jmath` を使います:

i `\imath` j `\jmath` \hat{j} `\hat{\jmath}`

アラビア数字 0-9 もクラス 0 です. デフォルトのフォントは立体/ローマン体です.

0 1 2 3 4 5 6 7 8 9

3.3.2. ギリシア文字

ラテン文字のように, ギリシャ文字は簡単な記号, クラス 0 です. あいまいな歴史的な理由から, 数式の小文字のギリシャ文字のデフォルトフォントはイタリックで, 大文字のギリシャ文字のデフォルトフォントは立体/ローマン体です. しかし, 物理学や化学などの他の分野と, 伝統的な数学の文字の使い方は多少異なります. このリストにはないギリシャ語の文字は, アルファの場合は A, ベータの場合は B, 等々. 小文字のリストには, 外観がラテン文字の o と同じであるため, オミクロンはありません. 実際には, 混乱を避けるために, ラテン語と似ているギリシャ文字は, 数式にほとんど使われていません.

Γ <code>\Gamma</code>	α <code>\alpha</code>	ν <code>\nu</code>	F <code>\digamma</code>
Δ <code>\Delta</code>	β <code>\beta</code>	ξ <code>\xi</code>	ε <code>\varepsilon</code>
Λ <code>\Lambda</code>	γ <code>\gamma</code>	π <code>\pi</code>	\varkappa <code>\varkappa</code>
Φ <code>\Phi</code>	δ <code>\delta</code>	ρ <code>\rho</code>	φ <code>\varphi</code>
Π <code>\Pi</code>	ϵ <code>\epsilon</code>	σ <code>\sigma</code>	ϖ <code>\varpi</code>
Ψ <code>\Psi</code>	ζ <code>\zeta</code>	τ <code>\tau</code>	ϱ <code>\varrho</code>
Σ <code>\Sigma</code>	η <code>\eta</code>	υ <code>\upsilon</code>	ς <code>\varsigma</code>
Θ <code>\Theta</code>	θ <code>\theta</code>	ϕ <code>\phi</code>	ϑ <code>\vartheta</code>
Υ <code>\Upsilon</code>	ι <code>\iota</code>	χ <code>\chi</code>	
Ξ <code>\Xi</code>	κ <code>\kappa</code>	ψ <code>\psi</code>	
Ω <code>\Omega</code>	λ <code>\lambda</code>	ω <code>\omega</code>	
	μ <code>\mu</code>		

3.3.3. その他の“基本”アルファベット記号

これらもクラス 0 です.

\aleph^a <code>\aleph^a</code>	ℓ <code>\ell</code>	∂ <code>\partial</code>	\Game <code>\Game</code>
\beth <code>\beth</code>	\eth <code>\eth</code>	\wp <code>\wp</code>	\Im <code>\Im</code>
\daleth <code>\daleth</code>	\hbar <code>\hbar</code>	\S <code>\S</code>	\Re <code>\Re</code>
\gimel <code>\gimel</code>	\hslash <code>\hslash</code>	\Bbbk <code>\Bbbk</code>	
\complement <code>\complement</code>	\mho <code>\mho</code>	\Finv <code>\Finv</code>	

注意 1. ラベル^{a,b}がつけられてるフォントは `amssymb` が読み込まれたときの, `msam` あるいは `msbm` のことです.

3.3.4. 数式フォントの変更

包括的な数学フォントの変更をサポートするために必要なフォントのすべてが、通常の \LaTeX 設定で一般的に利用できるわけではありません。ここでは、Computer Modern フォントの標準セットが使用されているときに、幅広い数学記号にさまざまなフォント変換を適用した結果があります。すべてのフォント変更に正しく応答する唯一の記号は大文字のラテン文字であることがわかります。実際、ラテン文字とは別に**ほぼすべての数学記号**はフォントの変更に影響されません。小文字のラテン文字、大文字のギリシャ文字、数字は一部のフォント変更に正しく対応しますが、他のフォント変更では奇妙な結果が生じます。(Lucida New Math などの数学フォントセットを使用すると、状況が多少改善される可能性があります)。

default	\mathbf	\mathrm	\mathsf	\mathit	\mathcal	\mathbb	\mathfrak
X	\mathbf{X}	X	X	X	\mathcal{X}	\mathbb{X}	\mathfrak{X}
x	\mathbf{x}	x	x	x	\mathcal{x}	\mathbb{x}	\mathfrak{x}
0	$\mathbf{0}$	$\mathrm{0}$	$\mathsf{0}$	$\mathit{0}$	$\mathcal{0}$	$\mathbb{0}$	$\mathfrak{0}$
$[]$	$\mathbf{[]}$	$\mathrm{[]}$	$\mathsf{[]}$	$\mathit{[]}$	$\mathcal{[]}$	$\mathbb{[]}$	$\mathfrak{[]}$
$+$	$\mathbf{+}$	$\mathrm{+}$	$\mathsf{+}$	$\mathit{+}$	$\mathcal{+}$	$\mathbb{+}$	$\mathfrak{+}$
$-$	$\mathbf{-}$	$\mathrm{-}$	$\mathsf{-}$	$\mathit{-}$	$\mathcal{-}$	$\mathbb{-}$	$\mathfrak{-}$
$=$	$\mathbf{=}$	$\mathrm{=}$	$\mathsf{=}$	$\mathit{=}$	$\mathcal{=}$	$\mathbb{=}$	$\mathfrak{=}$
Ξ	$\mathbf{\Xi}$	$\mathrm{\Xi}$	$\mathsf{\Xi}$	$\mathit{\Xi}$	$\mathcal{\Xi}$	$\mathbb{\Xi}$	$\mathfrak{\Xi}$
ξ	$\mathbf{\xi}$	$\mathrm{\xi}$	$\mathsf{\xi}$	$\mathit{\xi}$	$\mathcal{\xi}$	$\mathbb{\xi}$	$\mathfrak{\xi}$
∞	$\mathbf{\infty}$	$\mathrm{\infty}$	$\mathsf{\infty}$	$\mathit{\infty}$	$\mathcal{\infty}$	$\mathbb{\infty}$	$\mathfrak{\infty}$
\aleph	$\mathbf{\aleph}$	$\mathrm{\aleph}$	$\mathsf{\aleph}$	$\mathit{\aleph}$	$\mathcal{\aleph}$	$\mathbb{\aleph}$	$\mathfrak{\aleph}$
\sum	$\mathbf{\sum}$	$\mathrm{\sum}$	$\mathsf{\sum}$	$\mathit{\sum}$	$\mathcal{\sum}$	$\mathbb{\sum}$	$\mathfrak{\sum}$
Π	$\mathbf{\Pi}$	$\mathrm{\Pi}$	$\mathsf{\Pi}$	$\mathit{\Pi}$	$\mathcal{\Pi}$	$\mathbb{\Pi}$	$\mathfrak{\Pi}$
\Re	$\mathbf{\Re}$	$\mathrm{\Re}$	$\mathsf{\Re}$	$\mathit{\Re}$	$\mathcal{\Re}$	$\mathbb{\Re}$	$\mathfrak{\Re}$

一般的な望みは、特定の数学記号の太字のバージョンを得ることです。 \mathbf が適用されない記号については、 \boldsymbol または \pmb コマンドを使用できます。

$$A_\infty + \pi A_0 \sim \mathbf{A}_\infty + \pi \mathbf{A}_0 \sim \mathbf{A}_\infty + \pi \mathbf{A}_0 \tag{3.1}$$

```
A_\infty + \pi A_0
\sim \mathbf{A}_{\boldsymbol{\infty}} \boldsymbol{+}
\boldsymbol{\pi} \mathbf{A}_{\boldsymbol{0}}
\sim \pmb{A}_{\pmb{\infty}} \pmb{+} \pmb{\pi} \pmb{A}_{\pmb{0}}
```

\boldsymbol コマンドは、 \amsmath パッケージよりも新しい、強力なバージョンである \bm パッケージを読み込んで使うのが望ましいです。一度に複数の記号に \boldsymbol を適用することは通常はお勧めできません。もしそのような必要性が生じているようであれば、別のより良い方法があることを意味します。

3.3.5. 黒板太字 (msbm; ただし小文字はありません)

使い方は $\text{\mathbb}{R}$ です。 \amsmath が必要です。

ABCDEFGHIJKLMNOPQRSTUVWXYZ

一つだけ小文字が用意されています: $\text{\kern 0pt \mathbb{k}}$

3.3.6. 飾り文字 (cmsy; 小文字はありません)

使い方: $\text{\mathcal}{M}$.

ABCDEFGHIJKLMN O PQRSTU VWXY Z

3.3.7. CM フォントではない飾り文字と手書き文字

(rsfs; 小文字はありません) 使い方: `\usepackage{mathrsfs} \mathscr{B}`.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

(eusm; 小文字はありません) 使い方: `\usepackage{euscript} \mathscr{E}`.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

3.3.8. フラクトゥール文字 (eufm)

使い方 `\mathfrak{S}`. `amsmath` が必要です.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

3.4. いくつかの単純な記号. これらの文字もクラス 0 (通常) です. つまり組み込みの空白がありません.

# \#	/ \diagup	¬ \neg
& \&	◇ \diamondsuit	∄ \nexists ^a
∠ \angle ^b	∅ \emptyset	/ \prime
\ \backprime	∃ \exists	‡ \sharp ^b
★ \bigstar ^a	♭ \flat ^b	♠ \spadesuit
◆ \blacklozenge	∀ \forall	◁ \sphericalangle ^b
■ \blacksquare	♥ \heartsuit	□ \square
▲ \blacktriangle ^a	∞ \infty	√ \surd
▼ \blacktriangledown ^a	◇ \lozenge	⊤ \top
⊥ \bot	∠ \measuredangle ^b	△ \triangle
♣ \clubsuit	∇ \nabla	▽ \triangledown ^a
\ \diagdown	‡ \natural ^b	∅ \varnothing

注意 1. ラベル^{a,b}がつけられてるフォントは `amssymb` が読み込まれたときの, `msam` あるいは `msbm` のことです.

注意 2. 記号 □ と # を使用する際のよくある間違いは, 適切に定義された数学記号コマンドを使用せずに, 2 項演算子または関係記号として用いることです. 単に既存のコマンド `\square` または `\#` を使用すると, これらのコマンドがクラス 0 記号を生成するため, 記号間の間隔が正しくありません.

注意 3. 同義語 `\lnot`

3.5. 二項演算の記号

* *	⊖ \circleddash ^a	⊕ \oplus
+ +	∪ \cup	⊗ \oslash
- -	⊍ \Cup ^a	⊗ \otimes
∏ \amalg	∿ \curlyvee ^a	± \pm
* \ast	⋈ \curlywedge ^a	⋈ \rightthreetimes ^a
⌋ \barwedge ^a	† \dagger	× \rtimes ^b
○ \bigcirc	‡ \ddagger	\ \setminusminus
▽ \bigtriangledown	◇ \diamond	⋈ \smallsetminusminus ^b
△ \bigtriangleup	÷ \div	□ \sqcap
◻ \boxdot ^a	∗ \divideontimes ^b	□ \sqcup
◻ \boxminus ^a	‡ \dotplus ^a	★ \star
◻ \boxplus ^a	⌋ \doublebarwedge ^a	× \times
⊠ \boxtimes ^a	⋈ \gtrdot ^b	◁ \triangleleft
• \bullet	⌋ \intercal ^a	▷ \triangleright
∩ \cap	⋈ \leftthreetimes ^a	⊕ \uplus
∩ \Cap ^a	⋈ \lessdot ^b	∨ \vee
⋈ \cdot	⋈ \ltimes ^b	∨ \veebar ^a
⋈ \centerdot ^a	⊎ \mp	∧ \wedge
○ \circ	⊙ \odot	⋈ \wr
⊗ \circledast ^a	⊖ \ominus	
⊖ \circledcirc ^a		

注意 1. ラベル^{a,b}がつけられてるフォントは `amssymb` が読み込まれたときの, `msam` あるいは `msbm` のことです.

同義語: `\land`, `\lor`, `\doublecup`, `\doublecap`

3.6. 関係記号: $< = > \succ \sim$ および, それらの変種

$<$	\geqslant \geqslant ^a	\lesssim \lnapprox ^b	\succcurlyeq \preccurlyeq ^a
$=$	\gg \ggg ^a	\lesseqgtr \lneq ^b	\preceq \preceq
$>$	\ggg \ggg ^a	\lesseqgtr \lneqq ^b	\preccurlyeq \preccurlyeq ^b
\approx \approx	\gtrsim \gnapprox ^b	\lesssim \lnsim ^b	\precneqq \precneqq ^b
\approx \approxeq ^b	\gtrsim \gneq ^b	\lesssim \lvertneqq ^b	\precnsim \precnsim ^b
\asymp \asymp	\gtrsim \gneqq ^b	\ncong \ncong ^b	\prec \prec
\backsim \backsim ^a	\gtrsim \gnsim ^b	\neq \neq	\risingdotseq \risingdotseq ^a
\backsimeq \backsimeq ^a	\gtrapprox \gtrapprox ^a	\ngeq \ngeq ^b	\sim \sim
\bumpeq \bumpeq ^a	\gtreqless \gtreqless ^a	\ngeqq \ngeqq ^b	\simeq \simeq
\Bumpeq \Bumpeq ^a	\gtreqqlless \gtreqqlless ^a	\ngeqslant \ngeqslant ^b	\succ \succ
\circeq \circeq ^a	\gttrless \gttrless ^a	\ngtr \ngtr ^b	\succapprox \succapprox ^b
\cong \cong	\gttrsim \gttrsim ^a	\nleq \nleq ^b	\succcurlyeq \succcurlyeq ^a
\curlyeqprec \curlyeqprec ^a	\gvertneqq \gvertneqq ^b	\nleqq \nleqq ^b	\succeq \succeq
\curlyeqsucc \curlyeqsucc ^a	\leq \leq	\nleqslant \nleqslant ^b	\succapprox \succapprox ^b
\doteq \doteq	\leqq \leqq ^a	\nless \nless ^b	\succneqq \succneqq ^b
\doteqdot \doteqdot ^a	\leqslant \leqslant ^a	\nprec \nprec ^b	\succnsim \succnsim ^b
\eqcirc \eqcirc ^a	\lesapprox \lesapprox ^a	\nprec \nprec ^b	\succsim \succsim ^a
\eqsim \eqsim ^b	\lesseqgtr \lesseqgtr ^a	\nprec \nprec ^b	\thickapprox \thickapprox ^b
\eqslantgtr \eqslantgtr ^a	\lesseqqgtr \lesseqqgtr ^a	\nsim \nsim ^b	\thicksim \thicksim ^b
\eqslantless \eqslantless ^a	\lessgtr \lessgtr ^a	\nsucc \nsucc ^b	\triangleq \triangleq ^a
\equiv \equiv	\lessssim \lessssim ^a	\nsucceq \nsucceq ^b	
\fallingdotseq \fallingdotseq ^a	\ll \ll	\prec \prec	
\geq \geq	\lll \lll ^a	\precapprox \precapprox ^b	
\geqq \geqq ^a			

注意 1. ラベル^{a,b}がつけられてるフォントは `amssymb` が読み込まれたときの, `msam` あるいは `msbm` のことです.

同義語: `\ne`, `\le`, `\ge`, `\doteq`, `\lllless`, `\gggtr`

3.7. 関係記号 : 矢印. 次も参照 4 節.

\circlearrowleft^a	\Lleftarrow^a	\nwarrow
\circlearrowright^a	\longleftarrow	\rightarrow
\curvearrowleft^b	\Longleftarrow	\Rightarrow
\curvearrowright^b	\longleftrightarrow	\rightarrowtail^a
\downdownarrows^a	\Longleftrightarrow	\rightharpoondown
\downharpoonleft^a	\longmapsto	\rightharpoonup^a
\downharpoonright^a	\longrightarrow	\rightleftarrows^a
\hookleftarrow	\Longrightarrow	\rightleftharpoons^a
\hookrightarrow	\looparrowleft^a	\rightrightarrows^a
\leftarrow	\looparrowright^a	\rightsquigarrow^a
\Leftarrow	\lsh^a	\Rightarrow
\leftarrowtail^a	\mapsto	\Rsh^a
\leftharpoondown	\multimap^a	\searrow
\leftharpoonup	\nLeftarrow^b	\swarrow
\leftleftarrows^a	\nLeftrightarrow^b	\twoheadleftarrow^a
\leftrightarrows	\nRightarrow^b	\twoheadrightarrow^a
\Leftrightarrow	\nearrow	\upharpoonleft^a
\leftrightharpoons^a	\nleftarrow^b	\upharpoonright^a
\leftrightharpoons^a	\nleftrightarrow^b	\upuparrows^a
\leftrightsquigarrow^a	\nrightarrow^b	

注意 1. ラベル^{a,b}がつけられてるフォントは `amssymb` が読み込まれたときの, `msam` あるいは `msbm` のことです.

同義語: `\gets`, `\to`, `\restriction`

3.8. 関係記号 : その他

\backepsilon^b	\ntriangleright^b	\subsetneq^b
\because^a	\ntrianglerighteq^b	\supset
\between^a	\nvdash^b	\Supset^a
\blacktriangleleft^a	\nVdash^b	\supseteq
\blacktriangleright^a	\nvDash^b	\supseteqq^a
\bowtie	\nVDash^b	\supsetneq^b
\dashv	\parallel	\supsetneqq^b
\frown	\perp	\therefore^a
\in	\pitchfork^a	\trianglelefteq^a
\mid	\propto	\trianglerighteq^a
\models	\shortmid^b	\varpropto^a
\ni	\shortparallel^b	\varsubsetneq^b
\nmid^b	\smallfrown^a	\varsubsetneqq^b
\notin	\smallsmile^a	\varsupsetneq^b
\nparallel^b	\smile	\varsupsetneqq^b
\nshortmid^b	\sqsubset^a	\vartriangle^a
\nshortparallel^b	\sqsubseteq	\vartriangleleft^a
\nsupseteq^b	\sqsupseteq^a	\vartriangleright^a
\nsupseteqq^b	\subset	\vdash
\nsupseteq^b	\Subset^a	\Vdash^a
\nsupseteqq^b	\subseteq	\vDash^a
\ntriangleleft^b	\subseteqeq^a	\Vvdash^a
\ntrianglelefteq^b	\subsetneq^b	

注意 1. ラベル^{a,b}がつけられてるフォントは `amssymb` が読み込まれたときの, `msam` あるいは `msbm` のことです.

同義語: `\owns`

3.9. 大きさの変わる作用素

\int <code>\int</code>	\odot <code>\bigodot</code>	\uplus <code>\biguplus</code>	\prod <code>\prod</code>
\oint <code>\oint</code>	\oplus <code>\bigoplus</code>	\vee <code>\bigvee</code>	\int <code>\smallint</code>
\bigcap <code>\bigcap</code>	\otimes <code>\bigotimes</code>	\wedge <code>\bigwedge</code>	\sum <code>\sum</code>
\bigcup <code>\bigcup</code>	\sqcup <code>\bigsqcup</code>	\coprod <code>\coprod</code>	

3.10. 句読点

\dots	$::$	<code>\dotsm</code>
$//$	$!!$	<code>\dotso</code>
$ $	$??$	<code>\ddots</code>
$,,$	<code>\dotsb</code>	\vdots <code>\vdots</code>
$;;$	<code>\dotsc</code>	\dotsc <code>\dotsc</code>
$:$ <code>\colon</code>	<code>\dotsi</code>	

注意 1. $:$ はそれ自身でクラス 3 (関係) の空白をもつコロンを生成します。コロンのコマンドは、`f\colon A\to B` では $f: A \rightarrow B$ のように特別な空白を作成します。

注意 2. コマンド `\cdots` と `\ldots` は頻繁に使用されますが、意味に応じたコマンド `\dotsb` `\dotsc` `\dotsi` `\dotsm` `\dotso` の使用をお勧めします。(4.6 節をみてください)。

3.11. ペアになる (大きさの変わる) 記号. 詳しくは 6 節を参照のこと.

$()$ <code>()</code>	$\langle \rangle$ <code>\langle \rangle</code>
$[]$ <code>[]</code>	$\lceil \rceil$ <code>\lceil \rceil</code>
$\{ \}$ <code>\lbrace \rbrace</code>	$\lfloor \rfloor$ <code>\lfloor \rfloor</code>
$ $ <code>\lvert \rvert</code>	$\left(\right)$ <code>\lgroup \rgroup</code>
$ $ <code>\lVert \rVert</code>	$\left(\right)$ <code>\lmoustache \rmoustache</code>

3.12. ペアでは使われないが大きさの変わる記号

\lvert <code>\lvert</code>	\backslash <code>\backslash</code>	\uparrow <code>\uparrow</code>	$\} \}$ <code>\bracevert</code>
\lVert <code>\lVert</code>	\backslash <code>\backslash</code>	\Uparrow <code>\Uparrow</code>	

注意 1. `\lvert`, `|`, `\lVert`, `\l` をペアのデリミタ (語の前後を区切る, あるいは行列や絶対値など要素を囲む記号) として使うのは勧めません。(6.2 節をみてください) 代わりに, 3.11 節にある一覧表にあるデリミタを使用します。

同義語: `|| \l`

3.13. 大きさの変わる垂直の矢印

\uparrow <code>\uparrow</code>	\downarrow <code>\downarrow</code>	\updownarrow <code>\updownarrow</code>
\Uparrow <code>\Uparrow</code>	\Downarrow <code>\Downarrow</code>	\Updownarrow <code>\Updownarrow</code>

3.14. 数式で使われるアクセント

\acute{x} \acute{x}	\bar{x} \bar{x}	\vec{x} \vec{x}	\mathring{x} \mathring{x}
\grave{x} \grave{x}	\breve{x} \breve{x}	\dot{x} \dot{x}	\widetilde{xxx} \widetilde{xxx}
\ddot{x} \ddot{x}	\check{x} \check{x}	\ddot{x} \ddot{x}	\widehat{xxx} \widehat{xxx}
\tilde{x} \tilde{x}	\hat{x} \hat{x}	$\overset{\cdot}{x}$ \overset{\cdot}{x}	

3.15. 名前付き作用素. これらの作用素は複数の文字による省略形で表されています.

arccos \arccos	gcd \gcd	Pr \Pr
arcsin \arcsin	hom \hom	projlim \projlim
arctan \arctan	inf \inf	sec \sec
arg \arg	injlim \injlim	sin \sin
cos \cos	ker \ker	sinh \sinh
cosh \cosh	lg \lg	sup \sup
cot \cot	lim \lim	tan \tan
coth \coth	lim inf \liminf	tanh \tanh
csc \csc	lim sup \limsup	\varinjlim \varinjlim
deg \deg	ln \ln	\varprojlim \varprojlim
det \det	log \log	\varliminf \varliminf
dim \dim	max \max	\varlimsup \varlimsup
exp \exp	min \min	

上記のリストにない追加の名前付き作用素を定義するには, \DeclareMathOperator コマンドを使用します. たとえば,

```
\DeclareMathOperator{\rank}{rank}
\DeclareMathOperator{\esssup}{ess\,sup}
```

このようにできます.

$$\begin{aligned} \backslashrank(x) & \text{rank}(x) \\ \backslashesssup(y,z) & \text{ess sup}(y,z) \end{aligned}$$

\DeclareMathOperator*にある星印は, ディスプレイ数式, sup や max などにある範囲を示す式を使う作用素を作成します.

このようなすでに定義されている名前付き作用素で問題がある場合 (たとえば, 論文の表題や要約に 1 つ使用する場合など), 直接使用できる別の形式があります.

$$\backslashoperatorname{\rank}(x) \rightarrow \text{rank}(x)$$

4. 記法

4.1. 上側と下側を飾る. これらは視覚的にはアクセントと似ていますが, 1 つの基になる記号だけに適用されるのではなく, 複数の記号にまたがります. 参照を便利にするために, ここでは \widetilde と \widehat を重複して, 数式アクセントの表に含めています.

\widetilde{xxx} \widetilde{xxx}	\overleftarrow{xxx} \overleftarrow{xxx}
\widehat{xxx} \widehat{xxx}	\underleftarrow{xxx} \underleftarrow{xxx}
\overline{xxx} \overline{xxx}	\overrightarrow{xxx} \overrightarrow{xxx}
\underline{xxx} \underline{xxx}	\underrightarrow{xxx} \underrightarrow{xxx}
\overbrace{xxx} \overbrace{xxx}	\overleftrightarrow{xxx} \overleftrightarrow{xxx}
\underbrace{xxx} \underbrace{xxx}	$\underleftrightarrow{xxx}$ \underleftrightarrow{xxx}

4.2. 長さが伸びる矢印. \xleftarrow と \xrightarrow は, 普通より広い下付き文字や上付き文字に対応するために自動的に拡張される矢印を生成します. これらのコマンドは, 1 つのオプションの引数 (添え字) と 1 つの必須の引数 (上付き文字, 空の場合もある) を取ります:

$$A \xleftarrow{n+\mu-1} B \xrightarrow{n+i-1} C \tag{4.1}$$

```
\xleftarrow{n+\mu-1}\quad \xrightarrow[T]{n\pm i-1}
```

4.3. 記号に別の記号を重ねる. 標準のアクセント (3.14 節) に加えて, `\overset` と `\underset` コマンドで, 基礎におく記号の上または下に他の記号を配置することができます. たとえば, `\overset{*}{X}` とすれば, X の上に上付き小文字の $*$ を置くので, X^* になります. 8.4 節の `\sideset` の説明も参照してください.

4.4. 行列. `pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix`, `Vmatrix` の環境はそれぞれ適切なデリミタ `()`, `[]`, `{}`, `||`, `|||` が組み込まれています. また, デリミタをもたない `matrix` 環境と, 左揃えまたは列指定の他のバリエーションを取得するために使用できる `array` 環境があります.

```
\begin{pmatrix}
\alpha& \beta^* \\
\gamma^*& \delta
\end{pmatrix}
```

本文中での使用に適した小さな行列を生成するには, `smallmatrix` 環境 (たとえば, $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$) とします. そうすると通常の行列よりも小さくなるので, 文章の行の大きさにあいます. この例は,

```
\bigl( \begin{smallmatrix}
a&b \\
c&d
\end{smallmatrix} \bigr)
```

としたものです. デフォルトでは, 行列内のすべての要素は水平方向に中央揃えされます. `mathtools` パッケージは, これ以外の位置揃えを容易にするすべての行列環境の星印バージョンを提供します. このパッケージはまた, 星印バージョンと星印なしバージョンの両方で並列名をもつ `smallmatrix` の囲まれた版を提供します.

行列の中に指定された数の列にまたがるドットの行を生成するには `\hdotsfor` を使います. たとえば, 4 列の行列の 2 番目の列の `\hdotsfor{3}` は, 最後の 3 つの列にドットだけの行がプリントされます

区分関数の定義には, `cases` 環境があります:

```
P_{r-j}=\begin{cases}
0& \text{if } r-j \text{ is odd}, \\
r! \cdot (-1)^{(r-j)/2} & \text{if } r-j \text{ is even}.
\end{cases}
```

`\text` と埋め込み数式の使用に注意してください.

注意 plain TeX 形式の `\matrix{... \cr ... \cr}` と関連コマンド `\pmatrix`, `\cases` は L^AT_EX では避けるべきです (これらは `amsmath` パッケージがロードされると無効になります).

4.5. 数学用の空白コマンド. `amsmath` パッケージが使われていると, これらの数学用の空白コマンドはすべて, 数学モードの内側でも外側でも使用できます.

省略形	コマンド	例	省略形	コマンド	例
	<code>no space</code>	34		<code>no space</code>	34
<code>\,</code>	<code>\thinspace</code>	34	<code>\!</code>	<code>\negthinspace</code>	34
<code>\:</code>	<code>\medspace</code>	34		<code>\negmedspace</code>	34
<code>\;</code>	<code>\thickspace</code>	34		<code>\negthickspace</code>	34
	<code>\quad</code>	3 4			
	<code>\qquad</code>	3 4			

数式での間隔をより細かく制御するには, `\mspace` と '数学単位 (math units)' を使用します. 1 つの数学単位, すなわち `mu` は `mu` に等しい. したがって, 負の半分 `\quad` を得るには, `\mspace{-9.0mu}` と書きます.

これ以外に L^AT_EX では, 与えられた断片の高さまたは幅に等しい空白を入れる 3 つのコマンドもあります:

例	結果
<code></code>	3つの X に等しい空白. 高さは.
<code>\hphantom{XXX}</code>	3つの X のに等しい空白高さは 0
<code>\vphantom{X}</code>	0 に等しい幅, 高さ = X の高さ

4.6. ドット. 種々の状況において優先される省略記号 (斜めまたは水平オンライン) については一般的な合意がありません. そのため好み問題と考えられるかもしれません. ほとんどの場合, 一般的な `\dots` を使用することができ, `amsmath` は AMS の規則に従った斜めドット (`\ldots`) の間の中心ドット (`\cdots`) を指定することができます. ドットの後に続くものがあいまいである場合, コマンドにより特定の形式が使用できます. しかし, `\ldots` や `\cdots` の代わりに, 次に示すような意味に応じたコマンドを使うべきです.

- `\dotsc` は “コンマに続くドット”
- `\dotsb` は “に黄砂要素あるいは二項関係の後のドット”
- `\dotsm` は “たくさんのドット”
- `\dotsi` は “積分の後のドット”
- `\dotso` は “それ以外” (上の例に該当しない場合)

著者は, その場に応じた規約に適合させるべきである. たとえば, 出版社の規則に応じた規則に従って提出する必要がありますここではアメリカの数学会の慣習に従います:

<p>We have the series A_1, A_2, \dotsc, the regional sum $A_1 + A_2 + \dotsb$, the orthogonal product $A_1 A_2 \dotsm$, and the infinite integral $\int \int \dotso$.</p>	<p>We have the series A_1, A_2, \dots, the regional sum $A_1 + A_2 + \dots$, the orthogonal product $A_1 A_2 \dots$, and the infinite integral</p> $\int_{A_1} \int_{A_2} \dots$
---	---

4.7. 分割させないダッシュ. `\nbreakdash` コマンドはハイフンまたはダッシュの後の改行を抑制します. たとえば, ‘pages 1–9’ とするためには `pages 1\nbreakdash--9` と書くことで, ハイフンと 9 のあいだで改行は起こりません. `\nbreakdash` を使用すれば, `p-adic` のような組み合わせのときの望ましくないハイフネーションを防ぐこともできます. 頻繁に使用する場合は, 略語を使用することを勧めます. たとえば

```
\newcommand{\p}{ $\$p\$$ \nbreakdash}% for "\p adic" ("p-adic")
\nbreakdash\textendash}% for "pages 1\Ndash 9"
% For "\n dimensional" ("n-dimensional"):
\nbreakdash-\hspace{0pt}}
```

最後の例は, ハイフンの後で改行を禁止する方法を示していますが, 次の語には通常のハイフネーションを許可しています. ハイフンの後ろにゼロ幅の空白を追加します.

4.8. 根号. `\sqrt` コマンドは平方根を生成します. 基数を指定するには, オプション引数を 1 つ与えます.

$$\sqrt{\frac{n}{n-1}} S, \quad \sqrt[3]{2}$$

4.9. 数式の枠囲み. `\boxed` コマンドは, 引数の周りを枠で囲みます. `\fbox` は引数が数式にのみ適用できます:

$$\boxed{\eta \leq C(\delta(\eta) + \Lambda_M(0, \delta))} \tag{4.2}$$

```
\boxed{\eta \leq C(\delta(\eta) + \Lambda_M(0, \delta))}
```

数式番号を含む方程式を囲む必要がある場合は, 状況に応じて難しい場合があります. AMS の著者のための FAQ にいくつかの提案があります. ページの赤で概説されている項目を参照してください https://www.ams.org/faq?faq_id=290.

5. 分数とそれに関連する話題

5.1. `\frac`, `\dfrac`, および `\tfrac` コマンド. `\frac` コマンドは, 2つの引数—分子と分母—を取り, それらを通常の分数の形式にタイプセットします. `\dfrac` や `\tfrac` を使えば L^AT_EX が想定する大きさの代わりに (`t` = テキストスタイル, `d` = ディスプレイスタイル) 適切な大きさになります.

$$\frac{1}{k} \log_2 c(f), \quad \frac{1}{k} \log_2 c(f), \quad \frac{1}{k} \log_2 c(f) \quad (5.1)$$

```
\begin{equation}
\frac{1}{k}\log_2 c(f),\quad\dfrac{1}{k}\log_2 c(f),
\quad\tfrac{1}{k}\log_2 c(f)
\end{equation}
```

$$\Re z = \frac{n\pi \frac{\theta + \psi}{2}}{\left(\frac{\theta + \psi}{2}\right)^2 + \left(\frac{1}{2} \log \left| \frac{B}{A} \right| \right)^2}. \quad (5.2)$$

```
\begin{equation}
\Re{z} = \frac{n\pi \dfrac{\theta + \psi}{2}}{\left(\dfrac{\theta + \psi}{2}\right)^2 + \left(\dfrac{1}{2} \log \left| \dfrac{B}{A} \right| \right)^2}.
\end{equation}
```

5.2. `\binom`, `\dbinom`, および `\tbinom` コマンド. $\binom{n}{k}$ のような 2 項式の場合には, `\binom`, `\dbinom`, `\tbinom` というコマンドを使います:

$$2^k - \binom{k}{1} 2^{k-1} + \binom{k}{2} 2^{k-2} \quad (5.3)$$

```
2^k-\binom{k}{1}2^{k-1}+\binom{k}{2}2^{k-2}
```

5.3. `\genfrac` コマンド. `\frac`, `\binom` の機能とその変種は次のとおりです. 一般化された分数コマンド `\genfrac` は 6 つの引数をとります. 最後の 2 つは, ‘`\frac`’ の分子と分母に対応します. 最初の 2 つはオプションのデリミタです (`\binom` 参照). 3 番目は分母分子の間の線の太さを決めます (`\binom` はこれを使って分母分子の間の線の太さを 0pt に設定, つまり表示しません). 4 番目の引数は数式のスタイルを決めます: 整数値 0–3 は, それぞれ `\displaystyle`, `\textstyle`, `\scriptstyle`, `\scriptscriptstyle` を選択します. 3 番目の引数が空の場合, 線の太さはデフォルトで “normal” になります.

```
\genfrac{left-delim}{right-delim}{thickness}
{mathstyle}{numerator}{denominator}
```

これを説明するために, ここで `\frac`, `\tfrac`, そして `\binom` が, どのように定義されているかをみます.

```
\newcommand{\frac}[2]{\genfrac{}{}{}{#1}{#2}}
\newcommand{\tfrac}[2]{\genfrac{}{}{1}{#1}{#2}}
\newcommand{\binom}[2]{\genfrac{()}{0pt}{}{#1}{#2}}
```

注意 技術的な理由から, L^AT_EX ドキュメントでプリミティブ分数コマンド `\over`, `\atop`, `\above` を使用することは推奨されません (例: https://www.ams.org/faq?faq_id=288, 赤で囲まれたエントリ).

5.4. 連分数. 連分数

$$\frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \dots}}} \tag{5.4}$$

は、次のようにします。

```
\frac{1}{\sqrt{2}+
\frac{1}{\sqrt{2}+
\frac{1}{\sqrt{2}+\dotsb
}}}
```

こうすれば、`\frac` をそのまま使うよりは綺麗に作成されます。任意の分子の左または右の配置は、`\frac` の代わりに `\frac[1]` または `\frac[r]` を使用して行います。

6. デリミタ

6.1. デリミタの大きさ. 指定しない限り、数式のデリミタは、含まれる数式の標準の高さのままです。より大きなサイズを得るには、接頭辞 `\big ...` (下記参照) を使用して特定のサイズを選択するか、または自動サイズ設定のために接頭辞 `\left` と `\right` を使用できます。

`\left` と `\right` で行われるデリミタの自動的な大きさ調整には2つの制限があります：第1に、最大の数式を包むのに十分な大きさのでりミタを機械的に適用し、その次がかなり大きいデリミタへと一足飛びに拡大したものになります。これは、指定されたデリミタのサイズが極端に大きすぎる式は、全体のサイズが大きくなることを意味します。通常のサイズのテキストでは、6pt (3pt の上端と下端) へ飛びます。デリミタのサイズを調整すべき状況が2つまたは3つあります。これらの調整は、次のコマンドを使用して行われます：

デリミタ サイズ	サイズを 指定しない	<code>\left</code> <code>\right</code>	<code>\bigl</code> <code>\bigr</code>	<code>\Bigl</code> <code>\Bigr</code>	<code>\biggl</code> <code>\biggr</code>	<code>\Biggl</code> <code>\Biggr</code>
Result	$(b)\left(\frac{c}{d}\right)$	$(b)\left(\frac{c}{d}\right)$	$(b)\left(\frac{c}{d}\right)$	$(b)\left(\frac{c}{d}\right)$	$(b)\left(\frac{c}{d}\right)$	$(b)\left(\frac{c}{d}\right)$

第1の種類調整は、総和記号などの値の範囲を示す累積作用素に対して行われます。`\left` と `\right` を使うとデリミタは通常は必要なものより大きくなるので、`Big` または `bigg` を使用する方が良い結果が得られます：

$$\left[\sum_i a_i \left| \sum_j x_{ij} \right|^p \right]^{1/p} \quad \text{見比べましょう} \quad \left[\sum_i a_i \left| \sum_j x_{ij} \right|^p \right]^{1/p}$$

```
\biggl[\sum_i a_i\Bigl|\lvert\sum_j x_{ij}\Bigr\rvert^p\biggr]^{1/p}
```

2番目の種類の状況はペアと成っているデリミタをいくつか入れ子にして使う場合です。ここでは、`\left` と `\right` はすべて同じサイズになります (包含された材料をカバーするのに十分です)。デリミタを少し大きくして入れ子の要素を見やすくします。

```
((a_1 b_1) - (a_2 b_2))((a_2 b_1) + (a_1 b_2)) \quad \text{見比べましょう} \quad ((a_1 b_1) - (a_2 b_2))((a_2 b_1) + (a_1 b_2))
\left((a_1 b_1) - (a_2 b_2)\right)
\left((a_2 b_1) + (a_1 b_2)\right)
%\quad\text{versus}\quad
\quad\text{見比べましょう}\quad
\bigl((a_1 b_1) - (a_2 b_2)\bigr)
\bigl((a_2 b_1) + (a_1 b_2)\bigr)
```

3番目の状況は、 $\left| \frac{b'}{a'} \right|$ のように、`\left` と `\right` は広すぎる行間を引き起こします。その場合、`\bigl` と `\bigr` を使用してデリミタを生成できます。基本となるサイズよりも大きい、通常の行間にくらべて違和感が少ないようにできます： $\left| \frac{b'}{a'} \right|$ 。

`mathtools` パッケージにはサイジングを簡素化できる `\DeclarePairedDelimiter` 機能があります。詳細については、パッケージのドキュメントを参照してください。

6.2. 縦棒の記法. `|`を用いて (絶対値のような) ペアになったデリミタを生成することは推奨されません. 記号の向きについてあいまいさがあり, まれに間違っただけの間隔が生じます. たとえば, `|k|=|-k|` は `|k| = |-k|` を生成し, `|\sin x|` は正しい `|\sin x|` の代わりに `|\sin x|` を生成します. `\lvert` を “left vert bar” に `\rvert` を “right vert bar” に使うと, この問題が回避されます. `\lvert -k\rvert` によって生成された `|-k|` と比べてみましょう. 二重バーの場合は, `\lVert`, `\rVert` という似た方法があります. 推奨されるプラクティスは, ドキュメントのプリアンブルに, 縦棒の記号のペアを使うさいの適切なコマンドを定義しておくことです:

```
\providecommand{\abs}[1]{\lvert#1\rvert}
\providecommand{\norm}[1]{\lVert#1\rVert}
```

こうすれば `\abs{z}` は $|z|$ を生成し, `\norm{v}` は $\|v\|$ を生成します.

7. `\text` コマンド

`\text` コマンドの主な用途は, ディスプレイ内に単語または短い文を入れることです. これは `\mbox` と似ていますが, `\mbox` と異なり, 添え字を使用しても自動的に添字の大きさとテキストが生成されます.

$$f_{[x_{i-1}, x_i]} \text{ is monotonic, } i = 1, \dots, c+1 \quad (7.1)$$

```
f_{[x_{i-1}, x_i]} \text{ is monotonic,}
\quad i = 1, \dots, c+1
```

7.1. `\mod` と, その仲間. コマンド `\mod`, `\bmod`, `\pmod`, `\pod` は, “mod” 表記についての特別な空白規則を扱います. `\mod` と `\pod` は, `\pmod` の変わった仲間ですが, これらのほうを好む人がいるので用意されています. `\mod` はカッコを省略し, `\pod` は “mod” を省略してカッコを保持します.

$$\gcd(n, m \bmod n); \quad x \equiv y \pmod b; \quad x \equiv y \bmod c; \quad x \equiv y \pmod d \quad (7.2)$$

```
\gcd(n,m\bmod n) ;\quad x\equiv y\pmod b
;\quad x\equiv y\bmod c ;\quad x\equiv y\pod d
```

8. 積分と和

8.1. 範囲指定の配置. 積分, 合計, および類似の記号で変数のとる範囲を示す式は, 規則と文脈に応じて, 基礎記号の側面またはその上下に配置されます. L^AT_EX は自動的にどちらか一方を選択するための規則をもち, ほとんどの場合結果は問題ありません. そうでない場合, 範囲を示す式の配置に影響を与えるために使用できる 3 つコマンドが L^AT_EX にはあります: `\limits`, `\nolimits`, `\displaylimits` です.

$$\int_{|x-x_z(t)|<X_0} z^6(t)\phi(x) \quad \text{と} \quad \int_{|x-x_z(t)|<X_0} z^6(t)\phi(x)$$

```
\int_{\abs{x-x_z(t)}<X_0} ... \int\limits_{\abs{x-x_z(t)}<X_0} ...
```

`\limits` コマンドは, 適用される基礎記号の直後に続く必要があります. その意味は次のとおりです: この記号の通常の規則にかかわらず, 範囲を示す文字の場合, 通常の上付き文字 (右上) 下付き (右下) の位置ではなくて, それらを記号の真上か真下に配置します. `\nolimits` は, その代わりに側面にずらすことを意味し, 新しい種類の基礎記号の定義に使用できます. `\displaylimits` は, `\sum` コマンドのように標準的な配置を使用することを意味します.

詳しくは [AMUG] にある, `intlimits` と `nosumlimits` をみてください.

8.2. 多重積分. `\iint`, `\iiint`, `\iiiiiint` は, テキストスタイルとディスプレイスタイルの両方で, それらの間隔をきちんと調整した複数の積分記号を与えます. `\idotsint` は, 同じ考えの拡張したもので, 2つの積分記号の間にドット入れます. 意味を明確にするために dx やその類似の記号の前には小さい空白 (`\,`) を入れるべきであることに注意してください.

$$\iint_A f(x,y) dx dy \quad \iiint_A f(x,y,z) dx dy dz \quad (8.1)$$

$$\iiiiiint_A f(w,x,y,z) dw dx dy dz \quad \int \cdots \int_A f(x_1, \dots, x_k) \quad (8.2)$$

```
\iint\limits_A f(x,y)\,dx\,dy\qquad\iiint\limits_A
f(x,y,z)\,dx\,dy\,dz\
\iiiiiint\limits_A
f(w,x,y,z)\,dw\,dx\,dy\,dz\qquad\idotsint\limits_A f(x_1,\dots,x_k)
```

8.3. 複数行の下付き文字または上付き文字. `\substack` コマンドは, 複数行の下付き文字または上付き文字を生成するために使用できます: たとえば

$$\sum_{\substack{0 \leq i \leq m \\ 0 < j < n}} P(i,j)$$

8.4. `\sideset` コマンド. 特別な目的のために, `\sideset` というコマンドもあります. これは \sum や \prod のような記号の上付き文字の位置に記号を配置するためです. 注意: `\sideset` コマンドは, 大きな演算子記号でのみ使用するように設計されています. 普通の記号では結果は信頼できません. `\sideset` を使用すると,

$$\sideset{n < k, \text{\textit{\$n\$ odd}}}{\sum}' nE_n$$

とできます.

余分な空のカッコのペアは, `\sideset` が大きな演算子の各隅に余分な記号を置くことができるという事実によって説明されます. 総乗記号の各隅に星印を置くには, 次のように入力します.

$$\sideset{_*^*}_{_*^*} \prod \quad \prod^*$$

9. 数式の構成部分の大きさを変える

数式内のフォントサイズを変更するための L^AT_EX のメカニズムは, 数式の外で使用されているものとはまったく異なります. `\large` や `\huge` のようなテキストコマンドのいずれかを使って数式内の何かを大きくしようとすると:

`\large \#`

次のようなメッセージが返されます.

```
Command \large invalid in math mode
```

そのような試みは, L^AT_EX の数学記号どのように扱うかということについて誤解していることを示しています. 日常の文章で総和を表すのに # 記号を使うことがありますが, このように表示させるための最良の方法は, 標準の L^AT_EX `\DeclareMathSymbol` コマンドで “mathop” 型の記号として定義することです [LFG] を参照してください. (しかしこれは, 適切な文字サイズのスラッシュの表示サイズのペアで数学フォントを取得することを必要としますが, これはあまり簡単ではありません).

次の表示を考えてみましょう：

$$\frac{\sum_{n>0} z^n}{\prod_{1\leq k\leq n} (1-q^k)} \quad \frac{\text{\frac{\sum_{n > 0} z^n}}{\text{\prod_{1\leq k\leq n} (1-q^k)}}}$$

`\frac` の代わりに `\dfrac` を使用しても、この場合は何も変更されません。総和と総乗の記号をフルサイズで表示するには、`\displaystyle` コマンドが必要です：

$$\frac{\sum_{n>0} z^n}{\prod_{1\leq k\leq n} (1-q^k)} \quad \frac{\text{\displaystyle\sum_{n > 0} z^n}}{\text{\displaystyle\prod_{1\leq k\leq n} (1-q^k)}}$$

フルサイズの記号を使用したいが範囲指定を横に置きたい場合は、`\nolimits` コマンドを使用します：

$$\frac{\sum_{n>0} z^n}{\prod_{1\leq k\leq n} (1-q^k)} \quad \frac{\text{\displaystyle\sum\nolimits_{n > 0} z^n}}{\text{\displaystyle\prod\nolimits_{1\leq k\leq n} (1-q^k)}}$$

同じような機能コマンド `\textstyle`, `\scriptstyle`, および `\scriptscriptstyle` があります。これらは L^AT_EX に記号のサイズと空白について、現在の文脈上は他のサイズを生成する場合でも、(それぞれ) インライン数式、一次添字、または二次添字のサイズを適用します。

注意： これらのコマンドは、L^AT_EX で “宣言 (declaration)” と呼ばれる特別なクラスのコマンドに属します。特に、コマンドの有効範囲を指定する中カッコの位置に注意してください：

正しい： `\displaystyle ...` 間違い： `\displaystyle{...}`

10. このほかのパッケージ

数式のためドキュメントの組版に便利な他の多くの L^AT_EX パッケージが CTAN (Comprehensive T_EX アーカイブネットワーク) から利用できます。いくつかのお勧めを紹介します：

mathtools `amsmath` に、さらに機能を追加したもの：`amsmath` を読み込みます。

amsthm 定理と証明を作成するためのもの。

amsfonts `\mathbb` と `\mathfrak` を定義し、多くの記号を使うことができますようにします (`amssymb` に含まれています)。

accents アクセントと任意の記号を使ったアクセントの作成。

bm 太字のための数学パッケージで、`\boldsymbol` より一般的でより堅牢な実装を提供します。

mathrsfs Ralph Smith の筆記体です。

cases 2 つ以上の方程式のそれぞれに式番号を降らずに、大きなブレースカッコでまとめます。

delarray 配列の複数の行にまたがるデリミタです。

xypic 可換図式とそれ以外の図のため。

TikZ ダイアグラムをはじめ、多くのグラフ作成のため。

T_EX カタログの

<http://mirror.ctan.org/help/Catalogue/alpha.html>,
は、パッケージ名を知っている場合には適しています。

T_EX に関する質問や答えを調べる場合は、次のフォーラムがあります：

<https://tex.stackexchange.com/questions>

すでに、誰かによって答えが示されている場合があるので、まずアーカイブを調べます。整理された話題をもとに、そこから探すと答えを素早く得られます：

<https://tex.meta.stackexchange.com/a/2425#2425>

探しても見つからない場合には、あなたが初めての質問者となります。

11. その他の役に立つドキュメント

References

- [AMUG] American Mathematical Society and the \LaTeX 3 Project: *User's Guide for the `amsmath` package*, Version 2.+,
<http://mirror.ctan.org/macros/latex/required/amsmath/amslldoc.tex>
 and
<http://mirror.ctan.org/macros/latex/required/amsmath/amslldoc.pdf>,
 2017.
- [AFUG] American Mathematical Society: *User's Guide, AMSFonts*,
<http://mirror.ctan.org/fonts/amsmath/amslldoc.pdf>, 2002.
- [CLSL] Scott Pakin: *The Comprehensive \LaTeX Symbol List*,
<http://mirror.ctan.org/tex-archive/info/symbols/comprehensive/>,
 January 2017. Raw font tables, without symbol names, are shown alphabetically
 by font name in the `rawtables*.pdf` files in the same area of CTAN and from
 \TeX Live with `texdoc rawtables`.
- [Lam] Leslie Lamport: *\LaTeX : A document preparation system*, 2nd edition,
 Addison-Wesley, 1994.
- [LC] Frank Mittelbach and Michel Goossens, with Johannes Braams, David Carlisle,
 and Chris Rowley: *The \LaTeX Companion*, 2nd edition, Addison-Wesley, 2004.
- [LFG] \LaTeX 3 Project Team: *\LaTeX 2 _{ϵ} font selection*,
<http://mirror.ctan.org/macros/latex/doc/fntguide.pdf>, 2005.
- [LGC] Michel Goossens, Frank Mittelbach, Sebastian Rahtz, Denis Roegel, and
 Herbert Voß: *The \LaTeX Graphics Companion*, 2nd edition, Addison-Wesley,
 2008.
- [LGG] D. P. Carlisle, \LaTeX 3 Project: *Packages in the 'graphics' bundle*,
<http://mirror.ctan.org/macros/latex/required/graphics/grfguide.pdf>,
 2017.
- [LUG] \LaTeX 3 Project Team: *\LaTeX 2 _{ϵ} for authors*,
<http://mirror.ctan.org/macros/latex/doc/usrguide.pdf>, 2015.
- [MML] George Grätzer: *More Math into \LaTeX* , 5th edition, Springer, New York, 2016.
- [UCM] Will Robertson: *Every symbol (most symbols) defined by `unicode-math`*,
<http://mirror.ctan.org/macros/latex/contrib/unicode-math/unimath-symbols.pdf>, 2017; and
 Will Robertson, Philipp Stephani, Joseph Wright, and Khaled Hosny:
Experimental Unicode mathematical typesetting: The `unicode-math` package,
<http://mirror.ctan.org/macros/latex/contrib/unicode-math/unicode-math.pdf>, 2017.